**Relatório dos padrões de qualidade, práticas e procedimentos da <Sigla da estatal>**

**Controle de Versões**

*<Inserir os dados das versões.>*

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Data | Autor | Notas da Revisão |
|  |  |  |  |

**1. Introdução**

<descrever a função do relatório dos padrões de qualidade, práticas e procedimentos da qualidade e sua importância>

**2. Padrões de Qualidade, Práticas e Procedimentos**

<Relacionar os padrões de qualidade, práticas e procedimentos que a Estatal irá utilizar para controlar a qualidade sua qualidade >

|  |  |  |  |
| --- | --- | --- | --- |
| Código | Padrões de Qualidade | Práticas Adotadas | Procedimentos Adotados |
|  |  |  |  |
|  |  |  |  |

Aprovado em \_\_\_ de \_\_\_\_\_\_\_\_\_\_ de \_\_\_\_\_.

<nome completo da autoridade máxima da Estatal >

<cargo da autoridade máxima da Estatal >

**Observações:**

A finalidade deste relatório é fornecer um ponto de referência único sobre os padrões de qualidade, práticas e procedimentos adotados para o projeto. Trata-se de um relatório orientado a processo que destaca os elementos que contribuem para a obtenção dos objetivos de qualidade. O Plano de Garantia de Qualidade não conterá detalhes sobre as técnicas, os critérios, as métricas e outros itens das revisões e avaliações cujo enfoque seja o produto. Ele requer, por uma questão de conformidade a qualidade, que os detalhes da avaliação de qualidade do produto sejam fornecidos.

**Responsabilidade**

O Gerente de Projeto é responsável por garantir que o relatório seja criado, apropriado e aceitável para o projeto.

**Utilização de um Modelo de Qualidade**

Qualidade é estar em conformidade com os requisitos dos clientes. Qualidade é antecipar e satisfazer os desejos dos clientes. Confiança do cliente em estar obtendo um produto que atende a suas expectativas. Qualidade de produto de software deveria ser avaliada utilizando um modelo de qualidade definido. O modelo de qualidade deve ser usado quando se estabelecem metas de qualidade para produtos de software e produtos intermediários. A qualidade do produto de software deve ser hierarquicamente decomposta para um modelo de qualidade composto de características e sub características que podem ser usadas como um checklist de assuntos relacionados à qualidade. É praticamente impossível medir todas as sub características de todas as partes de um grande produto de software. Similarmente, não é prático medir qualidade em uso de todos os possíveis cenários de tarefas do usuário. Recursos para avaliação precisam ser alocados entre os diferentes tipos de medições dependendo dos objetivos do negócio e da natureza do produto e do processo. Agora será apresentado um modelo de qualidade para qualidade interna e externa, que categoriza a qualidade de software em 6 características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade):

**Funcionalidade**: Capacidade de fornecer funções que correspondam às necessidades explícitas e implícitas do usuário quando o software é utilizado sob condições especificadas.

* Adequação: capacidade de fornecer um conjunto apropriado de funções para tarefas específicas e objetivos do usuário.
* Acurácia: capacidade de fornecer o resultado com o grau de precisão desejado.
* Interoperabilidade: capacidade de interagir com um ou mais sistemas.
* Segurança de Acesso: capacidade de proteger dados e informações de pessoas ou sistemas não autorizados.
* Conformidade: capacidade de aderir a padrões, convenções, leis e prescrições similares relativas a funcionalidade.

**Confiabilidade:** Capacidade do software manter seu nível de desempenho quando utilizado em condições estabelecidas.

* Maturidade: capacidade de evitar defeitos no software.
* Tolerância a Falhas: capacidade de manter um nível de desempenho estabelecido em caso de defeito no software.
* Recuperabilidade: capacidade de recuperar dados diretamente afetados no caso de falhas.
* Conformidade: capacidade de aderir a padrões, convenções, leis e prescrições similares relativas a confiabilidade.

**Usabilidade:** Capacidade que o produto tem de ser entendido, aprendido, utilizado e ser atraente para o usuário.

* Inteligibilidade: capacidade do produto de fazer o usuário entender se o software é adequado, e como ele pode ser usado para tarefas particulares.
* Aprendibilidade: capacidade que o produto deve ter de fazer o usuário entendê-lo.
* Operacionalidade: capacidade que o produto deve ter para que o usuário possa aprendê-lo e controlá-lo.
* Atratividade: capacidade do produto em ser atraente para o usuário.
* Conformidade: capacidade de aderir a padrões, convenções, leis e prescrições similares relativas a usabilidade.

**Eficiência**: Relacionamento entre o nível de desempenho do software e a quantidade de recursos utilizados, sob condições estabelecidas.

* Comportamento em Relação ao Tempo: capacidade de fornecer tempos de resposta e processamento adequados, bem como taxas de transferência.
* Comportamento em Relação aos Recursos: capacidade de usar quantidade e tipos de recursos adequados.
* Conformidade: capacidade de aderir a padrões e convenções relativas a eficiência.

**Manutenibilidade**: Esforço necessário para se fazer modificações específicas no software.

* Analisabilidade: capacidade em diagnosticar deficiências e causas de defeitos.
* Modificabilidade: capacidade que o produto tem de receber modificações.
* Estabilidade: capacidade de evitar efeitos inesperados a partir de modificações.
* Testabilidade: capacidade de validar as modificações efetuadas no produto.
* Conformidade: capacidade de aderir a padrões e convenções relativas a manutenibilidade.

**Portabilidade**: Capacidade que o produto tem de ser transferido de um ambiente para outro.

* Adaptabilidade: capacidade de ser adaptado em diferentes ambientes sem intervenção.
* Capacidade de Instalação: capacidade de ser instalado em um ambiente específico.
* Coexistência: capacidade que o produto tem de coexistir com outro software independente em um ambiente comum, compartilhando recursos comuns.
* Capacidade de Substituição: capacidade que o produto de software deve ter de ser usado no lugar de outro produto de software com o mesmo propósito no mesmo ambiente.
* Conformidade: capacidade de aderir a padrões e convenções relativas a portabilidade.

Padrões de produto: – Se aplicam ao produto de software em desenvolvimento. Incluem padrões de documentos, como a estrutura de um documento de requisitos até como uma linguagem de programação deve ser usada. Padrões de processo: – Definem os processos que devem ser seguidos durante o desenvolvimento de software. São baseados no conhecimento sobre as melhores e mais apropriadas práticas para a empresa. Esse conhecimento frequentemente é adquirido somente após um grande número de tentativas e erros. Ajudam a empresa a evitar a repetição de erros cometidos no passado. Proveem um framework conceitual para a implementação do processo de garantia de qualidade. Considerando que esses padrões englobam as melhores práticas, a garantia da qualidade envolve assegurar que padrões apropriados foram selecionados e usados.

**Alguns Padrões de Qualidade de Software:**

* CMM – Capability Maturity Model – Desenvolvido pelo SEI (Software Engineering Institute);
* SPICE (Software Process Improvement & Capability dErtemination) – Desenvolvido pela “International Organization for Standardization and the International Electrotechnical Commission” (ISO/IEC) CMM.

Medições de software podem ser usadas para:

1. Fazer previsões gerais sobre um sistema
2. Identificar componentes anômalos

Controle – são associadas com processos de software. Exemplos:

* Tempo médio para reparar um defeito reportado;
* Tempo para rastrear um módulo.

Predição – são associadas com produtos de software. Exemplo: Complexidade de um módulo.

**Medidas de Produto:**

Dinâmicas: Coletadas por meio de medições realizadas em um programa em execução. Exemplo: Tempo para iniciar um programa.

Estáticas: Coletadas por meio de medições realizadas em representações do sistema, como projeto, código fonte ou documentação.